UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/613,824	07/03/2003	Mustafa Eroz	PD-203009	6911

7590   11/17/2005

Hughes Electronics Corporation
Patent Docket Administration
Bldg. 1, Mail Stop A109
P.O. Box 956
El Segundo, CA 90245-0956

EXAMINER
CHAUDRY, MUJTABA M

ART UNIT	PAPER NUMBER
2133	

DATE MAILED: 11/17/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

	Application No.	Applicant(s)
Office Action Summary	10/613,824	EROZ ET AL.
	Examiner	Art Unit	
	Mujtaba K. Chaudry	2133	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>03 July 2003</u>.

2a)☐ This action is **FINAL.**   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>1-22</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) <u>1-22</u> is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on <u>03 July 2003</u> is/are: a)☐ accepted or b)☒ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   a)☐ All   b)☐ Some * c)☐ None of:

    1.☐ Certified copies of the priority documents have been received.

    2.☐ Certified copies of the priority documents have been received in Application No. _____.

    3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

   * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date <u>2/27/2004</u>.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____ .

DETAILED ACTION

Information Disclosure Statement

The information disclosure statements (IDS) submitted on September 09, 2005 and

February 27, 2004 are in compliance with the provisions of 37 CFR 1.97 and have been

considered by the Examiner.

Oath/Declaration

The Oath filed July 03, 2003 complies with all the requirements set forth in MPEP 602 and

therefore is accepted.

Drawings

The drawings are objected to because:

-       Figure 1 should be designated by a legend such as --Prior Art-- because only that which is

old is illustrated. See MPEP § 608.02(g). Corrected drawings in compliance with 37

CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the

application. The replacement sheet(s) should be labeled "Replacement Sheet" in the page

header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If

the changes are not accepted by the examiner, the applicant will be notified and informed

of any required corrective action in the next Office action. The objection to the drawings

will not be held in abeyance.

- Figure 2 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). Corrected drawings in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

- Figure 3 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). Corrected drawings in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

- Figure 4 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). Corrected drawings in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed

of any required corrective action in the next Office action. The objection to the drawings

will not be held in abeyance.

- Figure 5 should be designated by a legend such as --Prior Art-- because only that which is

old is illustrated. See MPEP § 608.02(g). Corrected drawings in compliance with 37

CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the

application. The replacement sheet(s) should be labeled "Replacement Sheet" in the page

header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If

the changes are not accepted by the examiner, the applicant will be notified and informed

of any required corrective action in the next Office action. The objection to the drawings

will not be held in abeyance.

Appropriate correction is required.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or
> any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and
> requirements of this title.

Claims 1-10 are rejected under 35 U.S.C. 101 because the language of the claim raises a

questions as to whether the claim is directed merely to an abstract idea that is not tied to a

technological art, environment or machine which would result in a practical application

producing a concrete, useful, and tangible result to form the basis of statutory subject matter

under 35 USC 101.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claim 4 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing

to particularly point out and distinctly claim the subject matter which applicant regards as the

invention. The term "n" is not defined.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the

basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-3, 11 and 13 are rejected under 35 U.S.C. 102(e) as being anticipated by Richardson et

al. (USPN 6633856 B2).

As per claim 1, Richardson et al. (herein after referred to as one entity: Richardson)

teaches (cols. 2-3) LDPC codes are represented by bipartite graphs, often called Tanner graphs,

in which one set of nodes, the variable nodes, corresponds to bits of the codeword and the other

set of nodes, the constraint nodes, sometimes called check nodes, correspond to the set of parity-

check constraints which define the code. Edges in the graph connect variable nodes to constraint

nodes. A variable node and a constraint node are said to be neighbors if they are connected by an edge in the graph. For simplicity, we generally assume that a pair of nodes is connected by at most one edge. To each variable node is associated one bit of the codeword. A bit sequence associated one-to-one with the variable node sequence is a codeword of the code if and only if, for each constraint node, the bits neighboring the constraint (via their association with variable nodes) sum to zero modulo two, i.e., they comprise an even number of ones. The decoders and decoding algorithms used to decode LDPC codewords operate by exchanging messages within the graph along the edges and updating these messages by performing computations at the nodes based on the incoming messages. Such algorithms will be generally referred to as message passing algorithms. Each variable node in the graph is initially provided with a soft bit, termed a received value, that indicates an estimate of the associated bit's value as determined by observations from, e.g., the communications channel.

As per claim 2, Richardson teaches (Figure 15) an edge memory which stores the edge values as stated in the present application.
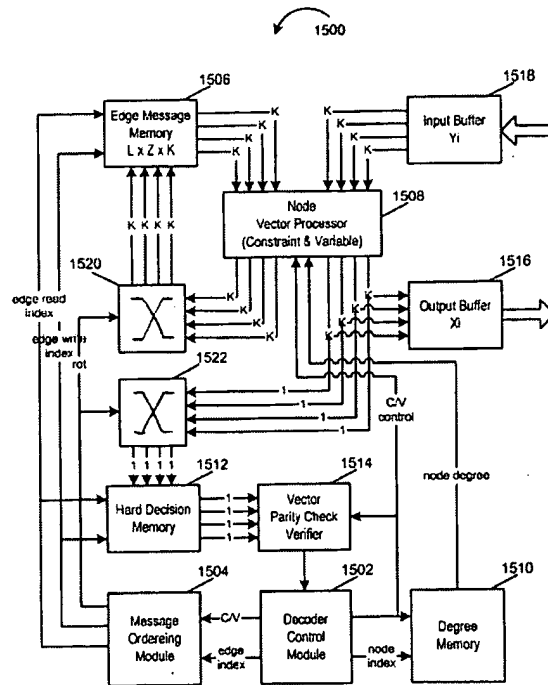
Figure 15

As per claim 3, Richardson teaches (Figure 15) the number of edges attached to a node, i.e., a variable node or constraint node, is referred to as the degree of the node. A regular graph or code is one for which all-variable nodes have the same degree, j say, and all constraint nodes have the same degree, k say. In this case we say that the code is a (j,k) regular code. In contrast to a "regular" code, an irregular code has constraint nodes and/or variable nodes of differing degrees. For example, some variable nodes may be of degree 4, others of degree 3 and still others of degree 2.

As per claim 11, Richardson teaches (cols. 2-3) LDPC codes are represented by bipartite graphs, often called Tanner graphs, in which one set of nodes, the variable nodes, corresponds to bits of the codeword and the other set of nodes, the constraint nodes, sometimes called check nodes, correspond to the set of parity-check constraints which define the code. Edges in the

graph connect variable nodes to constraint nodes. A variable node and a constraint node are said

to be neighbors if they are connected by an edge in the graph. For simplicity, we generally

assume that a pair of nodes is connected by at most one edge. To each variable node is

associated one bit of the codeword. A bit sequence associated one-to-one with the variable node

sequence is a codeword of the code if and only if, for each constraint node, the bits neighboring

the constraint (via their association with variable nodes) sum to zero modulo two, i.e., they

comprise an even number of ones. The decoders and decoding algorithms used to decode LDPC

codewords operate by exchanging messages within the graph along the edges and updating these

messages by performing computations at the nodes based on the incoming messages. Such

algorithms will be generally referred to as message passing algorithms. Each variable node in

the graph is initially provided with a soft bit, termed a received value, that indicates an estimate

of the associated bit's value as determined by observations from, e.g., the communications

channel. Richardson teaches (Figure 15) an edge memory which stores the edge values as stated

in the present application.

As per claim 13, Richardson teaches (Figure 15) the number of edges attached to a node,

i.e., a variable node or constraint node, is referred to as the degree of the node. A regular graph

or code is one for which all-variable nodes have the same degree, j say, and all constraint nodes

have the same degree, k say. In this case we say that the code is a (j,k) regular code. In contrast

to a "regular" code, an irregular code has constraint nodes and/or variable nodes of differing

degrees. For example, some variable nodes may be of degree 4, others of degree 3 and still

others of degree 2.

Claim Rejections - 35 USC § 103

The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966),

that are applied for establishing a background for determining obviousness under 35

U.S.C. 103(a) are summarized as follows:

1.   Determining the scope and contents of the prior art.
2.   Ascertaining the differences between the prior art and the claims at issue.
3.   Resolving the level of ordinary skill in the pertinent art.
4.   Considering objective evidence present in the application indicating obviousness
       or nonobviousness.

Claims 4-10, 12 and 14-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Richardson et al. (USPN 6633856 B2).

As per claim 4, Richardson substantially teaches, in view of above rejections, (col. 16 and

Figure 4) edge values 406 having n nodes and edge values having greater than n nodes 408.

Richardson does not explicitly teach to store the n edge values in a first portion and the

edge values greater than n in a second portion as stated in the present application.

However, in Figure 9, Richardson teaches that the edge values are overwritten after they

are processed by a node processor. Richardson teaches the decoder control unit 902 is

responsible for toggling the decoder operation between variable and check node processing

modes of operation, for determining when the iterative decoding process should be stopped, e.g.,

because of receipt of a convergence signal or reaching a maximum allowed iteration count, for

supplying or controlling the supply of degree information to the node processing unit and the

parity check verifier, and for controlling the supply of an edge index to the Message Ordering

Module 904. During operation, the decoder control module 902 transmits an edge index to the

message-ordering module 904. The value, edge index, is incremented over time to sequence

through all the edges in the graph. A different, e.g., unique, edge index is used for each edge in a

graph being implemented. In response to each received edge index, the message ordering

module will output an edge identifier, e.g., edge memory address information, thus selecting the

edge memory location that will be accessed, e.g., read from or written to, at any given time.

Assuming variable socket ordering, the message ordering module 904 will cause messages to be

read out and written back in sequential order during variable node processing and will cause the

messages to be read out and written back in order corresponding to constraint socket ordering

during constraint node processing. Therefore it would have been obvious to one of ordinary skill

in the art at the time the invention was made to store the n edge values in a first portion and the

edge values greater than n in a second portion within the decoding process of Richardson. This

modification would have been obvious to one of ordinary skill in the art because one of ordinary

skill in the art would have recognized that by storing the n edge values in a first portion and the

edge values greater than n in a second portion would have organizational complexity of the edge

memory storing the memory and therefore would have made it more accessible.

As per claim 5, Richardson substantially teaches (Figure 8) the relationship between the

12 edges of FIG. 6, as enumerated from the variable node side, in relationship to the variable and

check nodes to which they are connected. Row 802 shows the 5 variable nodes V.sub.1 through

V.sub.5. Beneath the variables 802 are shown the edges 1 through 12804 corresponding to the

associated sockets which are connected to the particular variable node. Note that since the edges

are ordered from the variable node side, in row 804 they appear in order from 1-12. During

variable node processing, the 12 edge messages in memory are accessed in sequence, e.g., in the

order shown in 804. Thus, during variable node processing, the messages may simply be read

out in order and supplied to a processing unit. The Examiner would like to point out that the edge

messages are accessed from the memory, which could be a read-only memory. This is an

obvious engineering design choice and does not necessarily change the overall operation of the

device.

As per claim 6, Richardson substantially teaches (col. 16) a serial LDPC decoder 900

which performs message processing operations sequentially, one edge at a time is shown in

Figure 9 and decoding using the exemplary code shown in Figure 6. The LDPC decoder 900

comprises a decoder control module 902, a message ordering module (socket permutation

memory) 904, a node degree memory 910, an edge memory 906, a node processor 908, output

buffer 916, hard decision memory 912 and parity check verifier 914.

As per claim 7, Richardson substantially teaches (Figure 4) the contiguous placement of

edges. The Examiner would like to point out that this would inherently alter and pose restrictions

on the parity check matrix.

As per claim 8, Richardson substantially teaches decoding low-density parity codes. The

Examiner would like to point out that it is well known in the art to use different modulation

schemes such as 8PSK, 16QAM, 16APSK, 32APSK and QPSK.

As per claim 9, Richardson substantially teaches (Figure 4) the edge values 404 are of a

fixed size.

As per claim 10, Richardson substantially teaches (col. 14) software LDPC decoder

implementations are possible wherein software is used to control a CPU to operate as a vector-

processing unit and to control passing of messages using a memory coupled to the CPU. In

software implementations, a single memory can also be used to store the decoder graph

description, edge messages as well as decoder routines used to control the CPU.

As per claim 12, Richardson substantially teaches (Figure 4) the contiguous placement of

edges. The Examiner would like to point out that this would inherently alter and pose restrictions

on the parity check matrix.

As per claim 14, Richardson substantially teaches, in view of above rejections, (col. 16

and Figure 4) edge values 406 having n nodes and edge values having greater than n nodes 408.

Richardson does not explicitly teach to store the n edge values in a first portion and the

edge values greater than n in a second portion as stated in the present application.

However, in Figure 9, Richardson teaches that the edge values are overwritten after they

are processed by a node processor. Richardson teaches the decoder control unit 902 is

responsible for toggling the decoder operation between variable and check node processing

modes of operation, for determining when the iterative decoding process should be stopped, e.g.,

because of receipt of a convergence signal or reaching a maximum allowed iteration count, for

supplying or controlling the supply of degree information to the node processing unit and the

parity check verifier, and for controlling the supply of an edge index to the Message Ordering

Module 904. During operation, the decoder control module 902 transmits an edge index to the

message-ordering module 904. The value, edge index, is incremented over time to sequence

through all the edges in the graph. A different, e.g., unique, edge index is used for each edge in a

graph being implemented. In response to each received edge index, the message ordering

module will output an edge identifier, e.g., edge memory address information, thus selecting the

edge memory location that will be accessed, e.g., read from or written to, at any given time.

Assuming variable socket ordering, the message ordering module 904 will cause messages to be read out and written back in sequential order during variable node processing and will cause the messages to be read out and written back in order corresponding to constraint socket ordering during constraint node processing. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to store the n edge values in a first portion and the edge values greater than n in a second portion within the decoding process of Richardson. This modification would have been obvious to one of ordinary skill in the art because one of ordinary skill in the art would have recognized that by storing the n edge values in a first portion and the edge values greater than n in a second portion would have organizational complexity of the edge memory storing the memory and therefore would have made it more accessible.

As per claim 15, Richardson substantially teaches (Figure 4) the contiguous placement of edges. The Examiner would like to point out that this inherently alters and poses restrictions on the parity check matrix.

As per claim 16, Richardson substantially teaches decoding low-density parity codes. The Examiner would like to point out that it is well known in the art to use different modulation schemes such as 8PSK, 16QAM, 16APSK, 32APSK and QPSK.

As per claim 17, Richardson substantially teaches (Figure 8) the relationship between the 12 edges of FIG. 6, as enumerated from the variable node side, in relationship to the variable and check nodes to which they are connected. Row 802 shows the 5 variable nodes V.sub.1 through V.sub.5. Beneath the variables 802 are shown the edges 1 through 12804 corresponding to the associated sockets which are connected to the particular variable node. Note that since the edges are ordered from the variable node side, in row 804 they appear in order from 1-12. During

variable node processing, the 12 edge messages in memory are accessed in sequence, e.g., in the

order shown in 804. Thus, during variable node processing, the messages may simply be read

out in order and supplied to a processing unit. The Examiner would like to point out that the edge

messages are accessed from the memory, which could be a read-only memory. This is an

obvious engineering design choice and does not necessarily change the overall operation of the

device.

As per claim 18, Richardson substantially teaches (col. 16) a serial LDPC decoder 900

which performs message processing operations sequentially, one edge at a time is shown in

Figure 9 and decoding using the exemplary code shown in Figure 6. The LDPC decoder 900

comprises a decoder control module 902, a message ordering module (socket permutation

memory) 904, a node degree memory 910, an edge memory 906, a node processor 908, output

buffer 916, hard decision memory 912 and parity check verifier 914.

As per claim 19, Richardson substantially teaches (cols. 2-3) LDPC codes are represented

by bipartite graphs, often called Tanner graphs, in which one set of nodes, the variable nodes,

corresponds to bits of the codeword and the other set of nodes, the constraint nodes, sometimes

called check nodes, correspond to the set of parity-check constraints which define the code.

Edges in the graph connect variable nodes to constraint nodes. A variable node and a constraint

node are said to be neighbors if they are connected by an edge in the graph. For simplicity, we

generally assume that a pair of nodes is connected by at most one edge. To each variable node is

associated one bit of the codeword. A bit sequence associated one-to-one with the variable node

sequence is a codeword of the code if and only if, for each constraint node, the bits neighboring

the constraint (via their association with variable nodes) sum to zero modulo two, i.e., they

comprise an even number of ones. The decoders and decoding algorithms used to decode LDPC

codewords operate by exchanging messages within the graph along the edges and updating these

messages by performing computations at the nodes based on the incoming messages. Such

algorithms will be generally referred to as message passing algorithms. Each variable node in

the graph is initially provided with a soft bit, termed a received value, that indicates an estimate

of the associated bit's value as determined by observations from, e.g., the communications

channel. Richardson teaches (Figure 15) an edge memory which stores the edge values as stated

in the present application.

Richardson does not explicitly teach to store the n edge values in a first portion and the

edge values greater than n in a second portion as stated in the present application.

However, in Figure 9, Richardson teaches that the edge values are overwritten after they

are processed by a node processor. Richardson teaches the decoder control unit 902 is

responsible for toggling the decoder operation between variable and check node processing

modes of operation, for determining when the iterative decoding process should be stopped, e.g.,

because of receipt of a convergence signal or reaching a maximum allowed iteration count, for

supplying or controlling the supply of degree information to the node processing unit and the

parity check verifier, and for controlling the supply of an edge index to the Message Ordering

Module 904. During operation, the decoder control module 902 transmits an edge index to the

message-ordering module 904. The value, edge index, is incremented over time to sequence

through all the edges in the graph. A different, e.g., unique, edge index is used for each edge in a

graph being implemented. In response to each received edge index, the message ordering

module will output an edge identifier, e.g., edge memory address information, thus selecting the

edge memory location that will be accessed, e.g., read from or written to, at any given time. Assuming variable socket ordering, the message ordering module 904 will cause messages to be read out and written back in sequential order during variable node processing and will cause the messages to be read out and written back in order corresponding to constraint socket ordering during constraint node processing. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to store the n edge values in a first portion and the edge values greater than n in a second portion within the decoding process of Richardson. This modification would have been obvious to one of ordinary skill in the art because one of ordinary skill in the art would have recognized that by storing the n edge values in a first portion and the edge values greater than n in a second portion would have organizational complexity of the edge memory storing the memory and therefore would have made it more accessible.

As per claims 20 and 21, Richardson substantially teaches (Figure 4) the contiguous placement of edges. The Examiner would like to point out that this would inherently alter and pose restrictions on the parity check matrix.

As per claim 22, Richardson substantially teaches decoding low-density parity codes. The Examiner would like to point out that it is well known in the art to use different modulation schemes such as 8PSK, 16QAM, 16APSK, 32APSK and QPSK.
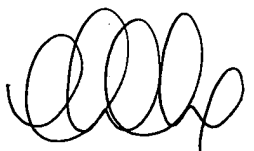
Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Additional pertinent prior arts are included herein for Applicant's review.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mujtaba K. Chaudry whose telephone number is 571-272-3817. The examiner can normally be reached on Mon-Thur 9-7:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Albert DeCady can be reached on 571-272-3819. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mujtaba Chaudry
Art Unit 2133
October 24, 2005

GUY LAMARRE
PRIMARY EXAMINER